

Diagrama de Classes

Sistemas UML

Faculdade GranTietê

Prof. José Antonio Gallo Junior

Introdução

O que é um Diagrama de Classes?

- Representação estrutural de um sistema.
- Define classes, atributos, métodos e relacionamentos.

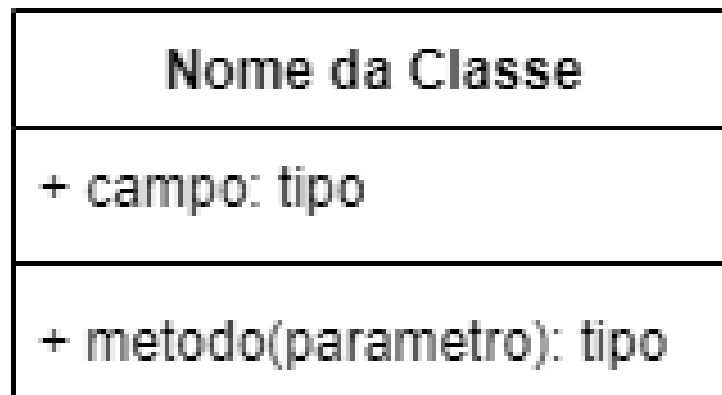
Por que usar?

- Organização do código antes da implementação.
- Melhora a comunicação entre desenvolvedores.
- Serve como documentação do sistema.

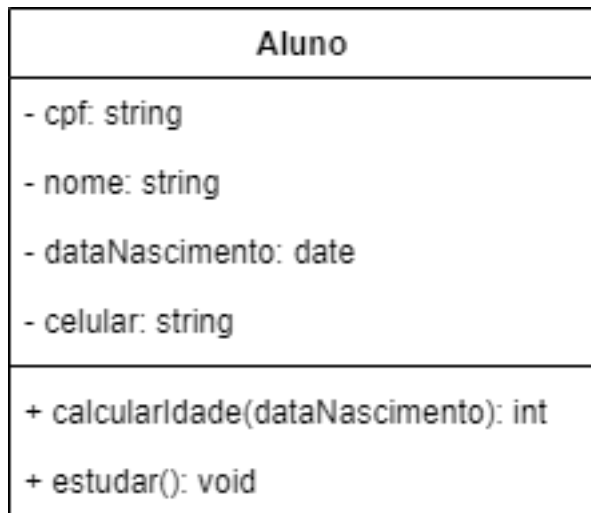
Estrutura de uma Classe na UML

Elementos básicos:

- Nome da classe
- Atributos (dados que descrevem a classe)
- Métodos (ações que a classe pode executar)



Estrutura de uma Classe na UML



Visibilidade:

- **"+" Público:** acessível de qualquer lugar.
- **"-" Privado:** acessível apenas dentro da classe.
- **"#" Protegido:** acessível na própria classe e em classes filhas.

Relacionamentos entre Classes

Tipos de Relacionamento:

- **Dependência:** Relacionamento simples.
- **Associação:** Ligação entre classes.
- **Agregação:** Uma classe pode existir sem a outra.
- **Composição:** Uma classe depende da outra para existir.
- **Generalização/Especialização:** Implementação da herança

Herança e Polimorfismo

Herança:

- Permite criar novas classes baseadas em outra.
- Evita repetição de código e melhora a organização.

Polimorfismo:

- Um método pode ter comportamentos diferentes em classes diferentes.
- **Exemplo:** O método fazerSom() pode ser diferente para Cachorro e Gato.

Interfaces e Dependências

Interfaces:

- Definem um contrato de métodos, mas não implementam código.

Dependências:

- Uma classe depende de outra para funcionar.
- Representada com setas tracejadas na UML.

Multiplicidade

Determina o número mínimo e máximo de objetos envolvidos na associação.

Multiplicidade	Significado
0..1	No mínimo zero e no máximo um. Indica não-obrigatoriedade do relacionamento
1..1	Um e somente um. Um objeto da classe se relaciona com um objeto de outra.
0..*	Mínimo zero e no máximo muitos.
1..*	Mínimo um e no máximo muitos.
*	Muitos.
2..7	Mínimo 2 e no máximo 7.

Relacionamento de Dependência

Dependência fraca, usualmente transiente, que ilustra que uma classe usa informações e serviços de outra classe em algum momento, dependendo dela.

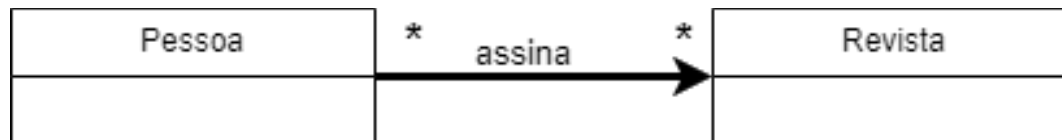
Do tipo “Classe A depende da Classe B”



Relacionamento de Associação

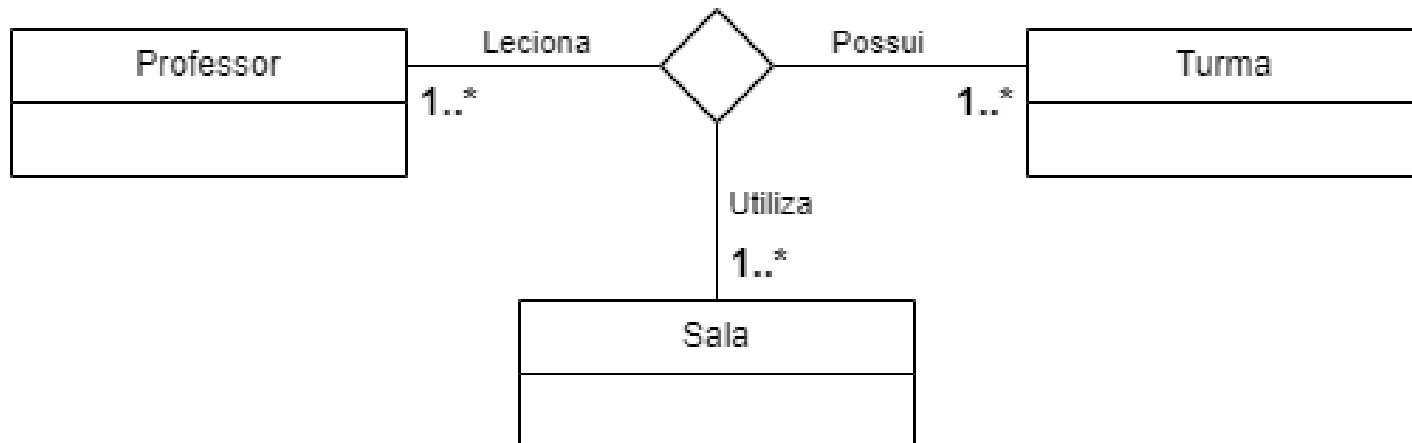
Relacionamento mais forte do que a dependência, indica que a classe mantém uma referência a outra classe ao longo do tempo. As associações podem conectar mais de duas classes.

Do tipo “Classe A tem uma Classe B”



Associação Ternária

Associação que conecta objetos de três classes. Um losango indica o ponto de conexão das classes envolvidas.



Relacionamento de Agregação

Relacionamento mais específico do que a associação, indica que uma classe é um contêiner ou uma coleção de outras classes. As classes contidas não dependem do contêiner – assim, quando o contêiner é destruído, as classes continuam existindo.

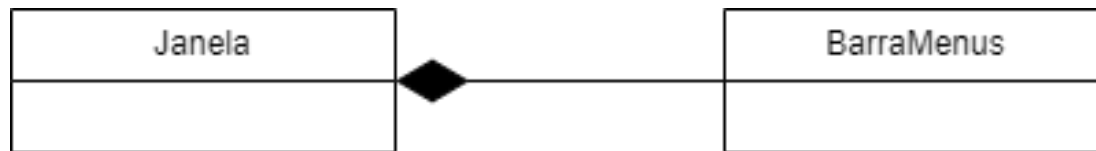
Do tipo “Classe A possui uma Classe B”



Relacionamento de Composição

Variação mais específico da agregação, este relacionamento indica uma dependência de ciclo de vida forte entre as classes, de modo que quando um contêiner é destruído, seu conteúdo também o é.

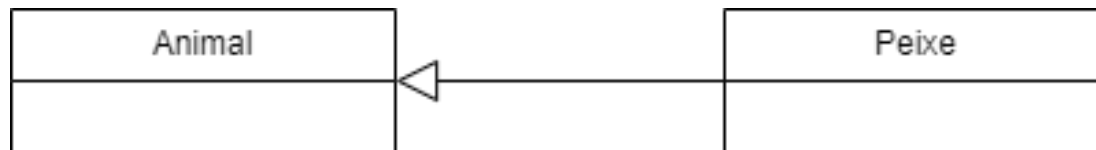
Do tipo “Classe A é parte da classe B”



Relacionamento de Generalização/Especialização

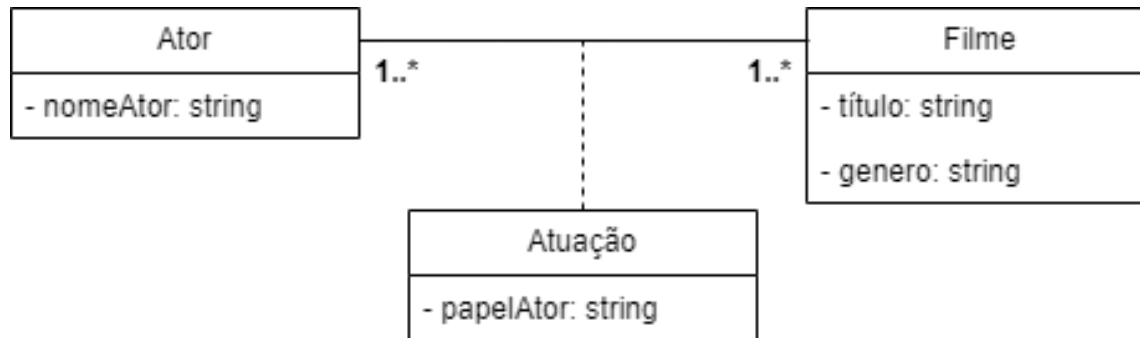
Relacionamento entre itens gerais (superclasses/classes-mãe) e tipos mais específicos desses itens (subclasses/classes-filhas). Representa a Herança entre as classes.

Do tipo “Classe A é um tipo de Classe B”

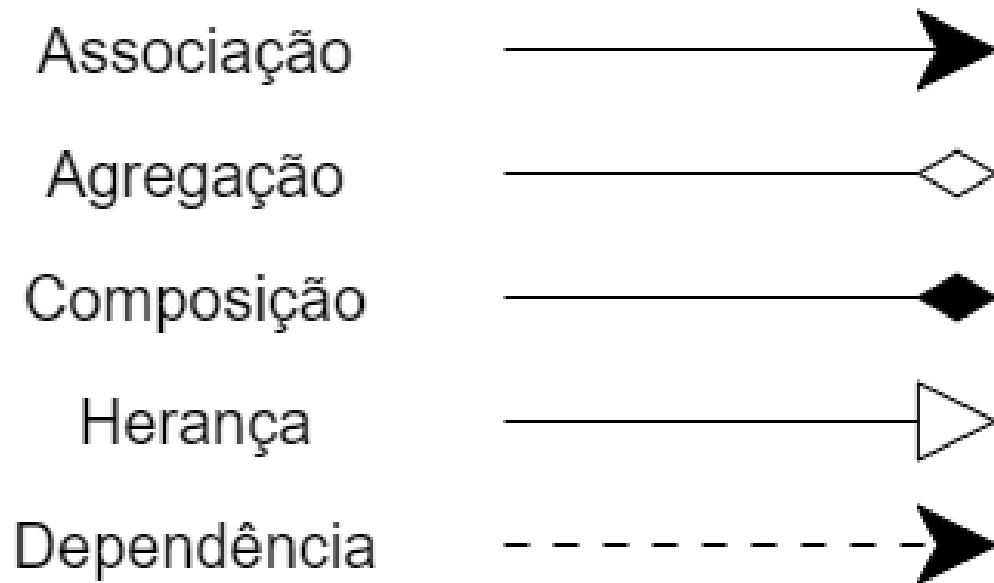


Classe Associativa

São produzidas quando ocorrem associações com multiplicidade muitos em todas as extremidades. No geral, existem atributos da associação que não podem ser armazenados em nenhuma das classes envolvidas.



Resumo da notação de Relacionamentos



Boas práticas

- O nome da classe deve ser significativo, descrevendo um aspecto real do sistema.
- Os relacionamentos entre os elementos devem ser identificados antes de criar o diagrama.
- Devem ser especificados os atributos e operações de cada classe.
- Sempre que necessário, acrescente anotações para ajudar a definir aspectos das classes ou seus relacionamentos